IAVO RESEARCH AND SCIENTIFIC
DBA of International Association of Virtual Organizations, Inc.
345 W. Main Street · Durham, NC 27701
919-433-2400

13 June 2008

Ms. Carolyn Sheaff
AFRL/IFEA
32 Brooks Road
Rome, NY 13441-4114

Ms. Sheaff:

Enclosed please find hardcopy of the Phase I Final Report, USAF Contract FA9550-07-C-0085 (re: *Platform Routing and Data Fusion Technologies for Cooperative ISR – fmCortex*^{TM}*)*.

This hardcopy satisfies the progress reporting requirements as per the provisions and instructions in the original Air Force contract (DD Form 1423 and Enclosure Number 1).

If you need to reach us for any reason, please contact Brad Grinstead (919-433-2401) or email (bgrinstead@iavo-rs.com).

Sincerely,

Matthew Heric
President

Enclosure

CC: Ms. Kathryn P. Pheulpin
AFOSR/PKR2
875 North Randolph Street, Rm 3112
Arlington, VA 22203

Dr. David Luginbuhl
AFOSR/NM
875 North Randolph Street, Rm 3112
Arlington, VA 22203

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to the Department of Defense, Executive Service Directorate (0704-0188). Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ORGANIZATION.

1. REPORT DATE *(DD-MM-YYYY)*
13-06-2008

2. REPORT TYPE
Final

3. DATES COVERED *(From - To)*

4. TITLE AND SUBTITLE
Platform Routing and Data Fusion Technologies for Cooperative ISR

5a. CONTRACT NUMBER
FA9550-07-C-0085

5b. GRANT NUMBER
AFOSR

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Dr. Brad Grinstead

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
IAVO Research and Scientific
(DBA for International Association of Virtual Organizations, Inc.)
345 W. Main Street, Suite 201
Durham, NC 27701

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AFOSR/NL
875 Randolph
Arlington VA 22203

10. SPONSOR/MONITOR'S ACRONYM(S)
AFOSR

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
AFRL-SR-AR-TR-08-0370

12. DISTRIBUTION/AVAILABILITY STATEMENT
Distribution A: Approved for Public Release

13. SUPPLEMENTARY NOTES

14. ABSTRACT
This report begins by providing a review of the original problem definition as well as the stated objectives for the Phase I research and development. Namely:

- Investigate optimal platform routing techniques and algorithms to optimize collection for fusion metric benefits while satisfying collection and de-confliction requirements;
- Select, research and define applications and appropriate optimization techniques for fusion of multi-source data from wide-body and UAV on-board sensors;
- Utilize Measures of Performance (MOPs) to determine fusion resultant improvements;

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:

a. REPORT	b. ABSTRACT	c. THIS PAGE

17. LIMITATION OF ABSTRACT

18. NUMBER OF PAGES

19a. NAME OF RESPONSIBLE PERSON

19b. TELEPHONE NUMBER *(Include area code)*

Reset

INSTRUCTIONS FOR COMPLETING SF 298

1. REPORT DATE. Full publication date, including day, month, if available. Must cite at least the year and be Year 2000 compliant, e.g. 30-06-1998; xx-06-1998; xx-xx-1998.

2. REPORT TYPE. State the type of report, such as final, technical, interim, memorandum, master's thesis, progress, quarterly, research, special, group study, etc.

3. DATES COVERED. Indicate the time during which the work was performed and the report was written, e.g., Jun 1997 - Jun 1998; 1-10 Jun 1996; May - Nov 1998; Nov 1998.

4. TITLE. Enter title and subtitle with volume number and part number, if applicable. On classified documents, enter the title classification in parentheses.

5a. CONTRACT NUMBER. Enter all contract numbers as they appear in the report, e.g. F33615-86-C-5169.

5b. GRANT NUMBER. Enter all grant numbers as they appear in the report, e.g. AFOSR-82-1234.

5c. PROGRAM ELEMENT NUMBER. Enter all program element numbers as they appear in the report, e.g. 61101A.

5d. PROJECT NUMBER. Enter all project numbers as they appear in the report, e.g. 1F665702D1257; ILIR.

5e. TASK NUMBER. Enter all task numbers as they appear in the report, e.g. 05; RF0330201; T4112.

5f. WORK UNIT NUMBER. Enter all work unit numbers as they appear in the report, e.g. 001; AFAPL30480105.

6. AUTHOR(S). Enter name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. The form of entry is the last name, first name, middle initial, and additional qualifiers separated by commas, e.g. Smith, Richard, J, Jr.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES). Self-explanatory.

8. PERFORMING ORGANIZATION REPORT NUMBER. Enter all unique alphanumeric report numbers assigned by the performing organization, e.g. BRL-1234; AFWL-TR-85-4017-Vol-21-PT-2.

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES). Enter the name and address of the organization(s) financially responsible for and monitoring the work.

10. SPONSOR/MONITOR'S ACRONYM(S). Enter, if available, e.g. BRL, ARDEC, NADC.

11. SPONSOR/MONITOR'S REPORT NUMBER(S). Enter report number as assigned by the sponsoring/ monitoring agency, if available, e.g. BRL-TR-829; -215.

12. DISTRIBUTION/AVAILABILITY STATEMENT. Use agency-mandated availability statements to indicate the public availability or distribution limitations of the report. If additional limitations/ restrictions or special markings are indicated, follow agency authorization procedures, e.g. RD/FRD, PROPIN, ITAR, etc. Include copyright information.

13. SUPPLEMENTARY NOTES. Enter information not included elsewhere such as: prepared in cooperation with; translation of; report supersedes; old edition number, etc.

14. ABSTRACT. A brief (approximately 200 words) factual summary of the most significant information.

15. SUBJECT TERMS. Key words or phrases identifying major concepts in the report.

16. SECURITY CLASSIFICATION. Enter security classification in accordance with security classification regulations, e.g. U, C, S, etc. If this form contains classified information, stamp classification level on the top and bottom of this page.

17. LIMITATION OF ABSTRACT. This block must be completed to assign a distribution limitation to the abstract. Enter UU (Unclassified Unlimited) or SAR (Same as Report). An entry in this block is necessary if the abstract is to be limited.

STTR Title: **Platform Routing and Data Fusion Technologies for Cooperative ISR – *fmCortex*^{TM}**

(AF07-T021)

Phase I Final Report

(Contract item number: 0001AC)

13 June 2008

Contract Number: FA9550-07-C-0085

Prepared by:

Dr. Brad Grinstead
IAVO Research and Scientific
(DBA for International Association of Virtual Organizations, Inc.)
345 W. Main Street, Suite 201
Durham, NC 27701
919-433-2401

Prepared for:

Ms. Carolyn Sheaff
AFRL/RIEA
32 Brooks Road
Rome, NY 13441-4114

Dr. David Luginbuhl
AFOSR/NM
875 North Randolph Street, Rm 3112
Arlington, VA 22203

TABLE OF CONTENTS

1 PROBLEM REVIEW

This report begins by providing a review of the original problem definition as well as the stated objectives for the Phase I research and development. Namely:

- Investigate optimal platform routing techniques and algorithms to optimize collection for fusion metric benefits while satisfying collection and de-confliction requirements;
- Select, research and define applications and appropriate optimization techniques for fusion of multi-source data from wide-body and UAV on-board sensors;
- Utilize Measures of Performance (MOPs) to determine fusion resultant improvements;
- Design an architecture to dynamically utilize the MOPs to enhance algorithm performance.

The significant growth in UAV platforms and payload capabilities offer a challenge and opportunity to realize the benefits of cooperative Intelligence, Surveillance, and Reconnaissance (ISR). The combination of both (1) wide-body and UAV cooperative systems or (2) multi-UAV (i.e., all-UAV) cooperative systems have positive, but different benefits to improved ISR. Among other technology challenges, the routing and data fusion requirements related to effective use of these cooperative platforms require unique algorithmic techniques and methods for evaluation of these innovative approaches.

Accordingly, our Phase I objectives were to:

- Define, design, and develop patterns for synergistic data fusion resource management techniques for effective, cooperative multi-platform ISR;
- Design, develop, and demonstrate an extended use case application of our baseline approach;
- Extend capabilities for combined geolocation, tracking, threat estimation and routing for modern hostile air defense environments; and, ultimately lay the foundation for the core ***fmCortex***^{TM} architecture (see Figure 1 and updated as shown in Figure 14).

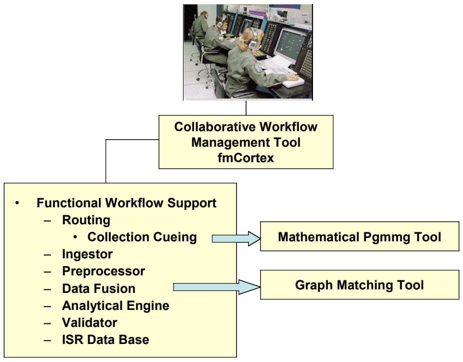


Figure 1. Proposed Conceptual ***fmCortex***^{TM} Architecture (Original).

To meet these objectives and thus address the defined problem we developed an approach that focused on a particular class of cooperative ISR missions that would benefit from improved technologies from both platform routing, data fusion as well as automated dynamic resource allocation techniques and automated support for inter-platform multi-operator workflow. The primary focus was on the combination of both cooperative-system wide-body and multi-UAV systems that provide positive and complementary benefits to improved, cooperative Intelligence, Surveillance, and Reconnaissance (ISR). Routing and data fusion requirements related to effective uses of these cooperative platforms demand progressive algorithmic techniques and methods embedded in theoretically innovative approaches - ***fmCortex***^{TM}.

2 PHASE I TASKING

An overarching objective of the ***fmCortex***^{TM} concept is to provide AFRL RIEA, Rivet Joint (RJ) and the CMS/KAST program a clear development path with respect to our ***fmCortex***^{TM} solution for use with legacy and future applications. Our intention was to address this with strict and constant consideration of the program office needs and requirements with the ultimate end-objectives of greatly improving operations tempo, and reducing program costs by laying the foundation for the development of a flexible and modularized end-to-end solution that could accommodate improvements while still allowing for technical migration as RJ needs dictate.

fmCortex^{TM} will ultimately include those pre- and post-processing issues critical for successful fusion (e.g., validation, standard-product assessment, entity/event/alert generation and re-generation, dynamic tasking, etc.), dynamic resource management, as well as dynamic database components. These will be characterized collectively as a *flow management* framework in which we will ultimately develop a modular design that accommodates future growth and continued research and development -- by either our team or others in the RJ community – via a componentized linearly sequential solution. Indeed, ***fmCortex***^{TM} is purposefully envisioned as having an open, modular, and extensible architecture that specifically allows and encourages future development and integration as collective RJ technical advances progress.

We undertook the Phase I with a set of eight specific tasks. Figure 2 illustrates these tasks and their nominal completion timelines.

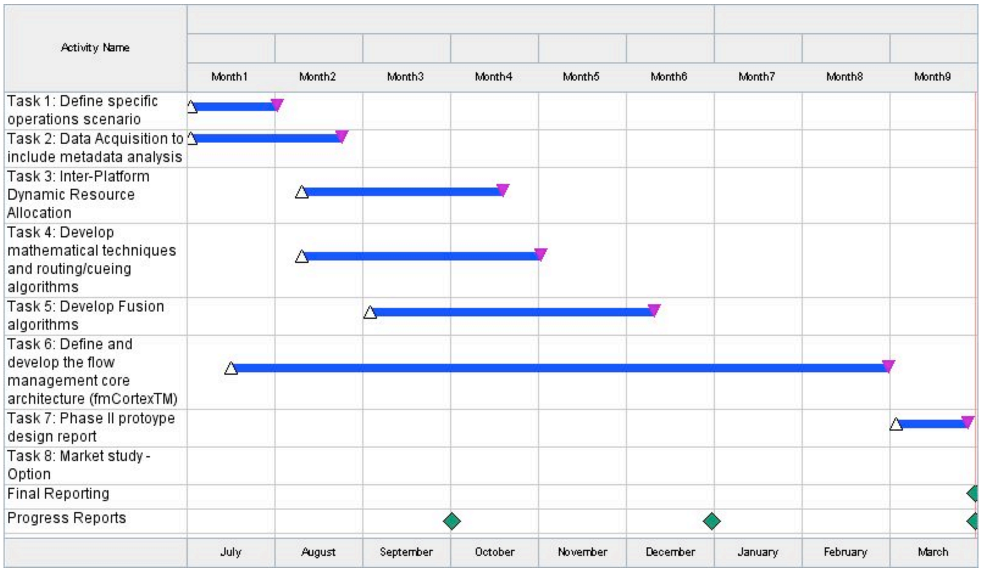


Figure 1. Phase I Schedule.

In the following sections we describe our Phase I accomplishment under each of the tasks as illustrated above.

2.1 TASK 1: DEFINE A SPECIFIC OPERATIONAL SCENARIO WITH EMPHASIS ON MISSION TASKING

Our baseline approach considered a dynamic adaptation to an existing ATO to deal with contingencies, pop-up threats, etc, while simultaneously finding the best way to reassign mission-tasking segments to the entire multiplatform suite. Our mathematical programming-based capability to optimize the task-to-platform reassignments allowed a solution to be framed that keeps overall mission effectiveness as its central optimization criterion, and can in fact be adaptively-implemented on a time-slice basis, so mission managers can specify and adjust optimization criteria in real-time. The Baseline Fusion approach involved both a “Level 1” and “Level 2” component addressing layered optimum emitter geolocation approach and emitter threat estimation logic. The Use Case involved a multiple-mission context and our demonstrated solution, while showing proof-of-concept, was a single solution at a point of time in the mission.

The Scenario, reported as a “Use Case” throughout the duration of our Phase I contract, involved a combined mission for regional IADS type surveillance and a simultaneous Special Operations mission that were interwoven to create the type of environment where Data fusion (DF) and Dynamic Resource Management (DRM) technologies would be emphasized. The scenario involved a pop-up emitter threat to a Special Operations (SpOps) mission that had to be serviced while balancing the regional surveillance requirement for the IADS.

Typical of the Cooperative ISR problem, there are interdependencies between the DF processes and the DRM resource and mission optimization techniques. The scenario geometry depicted in Figure 3 illustrates the platform general layouts.

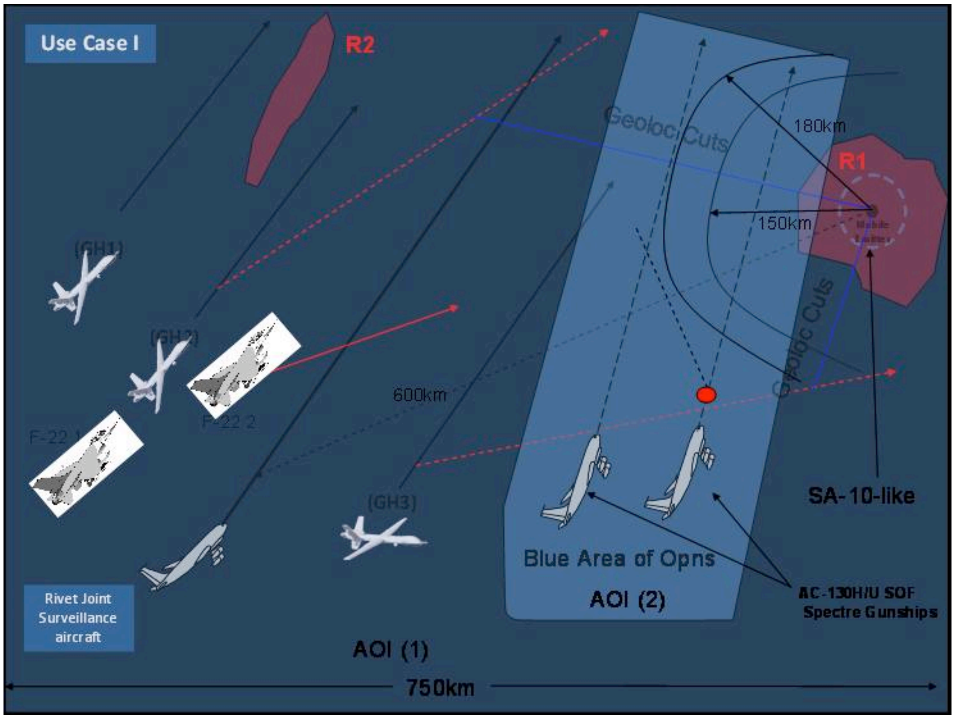


Figure 3. Use Case Geometry.

In the scenario, an RJ and three Global Hawk (GH) UAVs and an F-22 Strike aircraft are on a surveillance mission in area of interest *AOI (1)*; part of the mission is to over-fly region R2 where there are known threats. The F-22 Strike aircraft provide air cover for the sortie and the UAV’s are to provide specialized surveillance of certain priority sub-regions in R2. In a portion of *AOI (1)* labeled R1, two mobile SAM systems have been conducting harassing operations. R1 is defined by the boundaries of previously deployed locations of the emitters; the emitters are “SA-10-like”. The emitters have been performing irregular emission-no-emission cycling with occasional interspersed relocations. A synchronous SOF mission is also underway, involving squads of SOF warfighters being ferried by two AC-130 gunships into a target mission area to the Northeast of area of interest *AOI (2)*.

En-route the RJ notes one of the SA-10 emitters popping up somewhere in R1 (short on-air time) based on an AOA generally within the previously seen Red zone. However, the RJ-based and short-emitter uptime-based geolocation is not precise enough to fully understand the possible threat to the SOF platforms. The threat is sufficiently dangerous to the SOF mission in the *AOI (2)* area of operations that RJ issues a coded Alert to the SOF platforms and a request for UAV service to provide precision geolocation to better understand the SOF threat (this is Alert 1). In support of this request, pairwise UAV geolocation calculations are made along possible UAV trajectories that would fly the UAV's to tangent points on the emitter kill range circle. The Task Nomination Logic, as one basis to dynamically assign a particular pair of UAV's to the geolocation mission, uses these estimates. Other considerations in this optimization problem (see Section X for the details) involve the cost of degradation in the planned surveillance of R2 and other factors. A particular UAV pair is reassigned and begins flying the "tangent" trajectories. The UAV pair computes a series of geolocation updates according to the emitting/quiet time cycles of the emitter. At each calculation, the results are given to a threat estimation logic that computes "Time to Lethal Engagement (TTLE)" for the SOF platforms based on the estimated (fuzzy) time that it takes for the platforms to enter the kill zone of the SAM. Because of the uncertainty in the initial geolocation calculations, one F-22 is dynamically reassigned to provide air cover for the SOF platforms in case the kill zone envelope estimate is possibly in error. TTLE is also based on an estimate of the current SOF platform locations, which adds an error component to the overall geometry. The SOF platforms do not radiate and so the use of GPS location information for them is not available. The overall functional and processing flow is illustrated in Figure 4 which also describes the multiple geolocation algorithms that employ the sensing data from both the RJ and the UAV's, the dual multi-UAV Level 1 fusion geolocation algorithms, the fuzzy logic and fusion-based threat logic for Level 2/3 fusion, and the robust, multi-condition platform and functional reassignment logic.

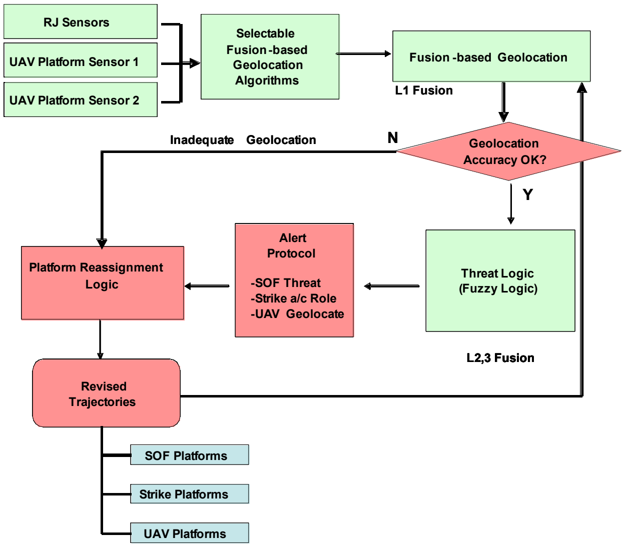


Figure 4. Overall Functional and Processing Flow.

2.1.1 Fusion-Based Geolocation Calculations

Phase (1) Calculations: Centralized Geolocation

As was described previously, the scenario begins with the RJ collecting some intermittent emissions from the harassing emitter in the R1 region; these data are used by organic RJ sensors to provide a rough geolocation. The RJ initializing an onboard Extended Kalman Filter (EKF) to estimate the absolute position of the unknown emitting target does this initial calculation. With the first sensed emission the RJ samples at discrete time intervals. Measurements fed to the EKF come from an RJ angle bearing sensor and ranging sensor. Both sensors are assumed to have statistical errors based on zero-mean Gaussian white noise.

Based on prior intelligence about region R1 we initialize the filter states somewhere within a broad initial R1 region with a large error covariance. The EKF algorithm onboard the RJ handles the irregular measurement data cycles of emission-no-emission coming from the harassing emitter. When measurement data is available during an emission cycle the EKF updates the estimated state and covariance. The EKF propagates the last state and covariance value forward at each time step in the no-emission cycle. Both state estimates errors and the error covariance increase over time during cycles of no emissions until a measurement is received.

It is assumed that the RJ gets a few "looks" at the emitter over a few emission cycles, and once the EKF converges, the RJ takes the emitter location estimate and error covariance and calculates a 3-sigma bound on the estimated emitter's geolocation, as shown in Figure 5. Values must be within a selectable threshold value with respect to the total region encompassing R1. A kill zone is added on the 3-sigma bound ellipse based on the prior intelligence of emitter types in region R1. Phase (1) final calculations are inadequate for accurate strike geolocation, yet critical for Phase (2) Error Covariance Reduction.

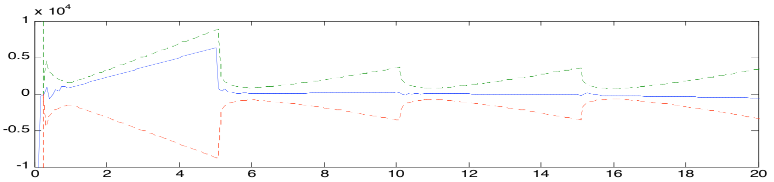


Figure 5. 3σ Bounds (Dashed) on the Geolocation Error (Solid).

Phase (2) Calculations: Propagated Centralized Geolocation – Error Covariance Reduction

Once the RJ Task Nomination Logic calculates the UAV trajectories with respect to the kill zone, three time-correlated positions along those trajectories are selected for each UAV. These three positions are taken along a virtual computed path towards the emitter but remaining outside the kill zone determined from the RJ solution, as shown in Figure 3; these are the "tangent" trajectories mentioned above. Using these three future positions for each UAV, the RJ computes the covariance of the expected errors of the geolocation solution using the UAVs. Propagated geolocation solutions depend now the UAV sensor geometry with respect to the estimated geolocation of the emitter. Note that the location of the emitter is required to compute this covariance. It will be assumed that the RJ-provided initial geolocation is "close enough" to provide a fairly accurate covariance analysis for each of the UAV-based solutions. Specifically it is assumed that the RJ initial geolocation errors only produce second-order errors in the computation of the forward time propagated UAV covariance analysis. A total of three solutions, containing emitter locations and associated error covariance matrices are determined. See figure 6.

UAV Pair-12	UAV Pair-23	UAV Pair-13
$P^{12} = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix}$	$P^{23} = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix}$	$P^{13} = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix}$
Normalized $P^{12} / p^{nm}{}_{ij(Max)}$	*Normalized* $P^{23} / p^{nm}{}_{ij(Max)}$	*Normalized* $P^{13} / P^{nm}{}_{ij(Max)}$

Where

n and m = 1, 2 & 3

i and j = 1 & 2

Table 1- Normalized UAV-Pair Team Performance

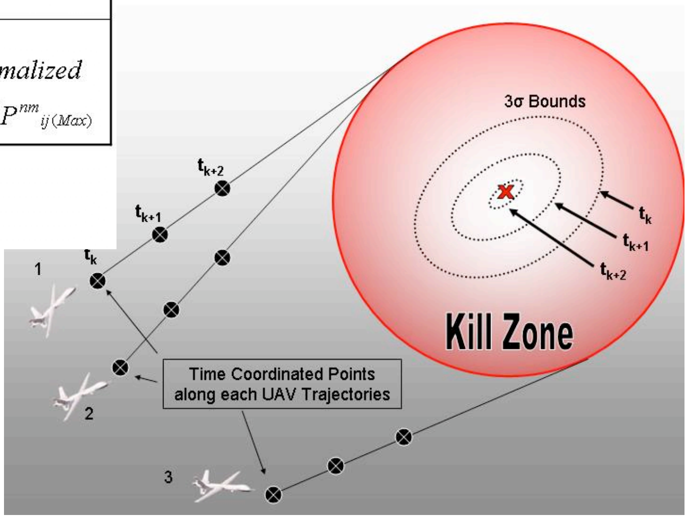


Figure 6. 3σ Error Bounds (dotted) Reduction Phase.

A simple weighting for each error covariance matrix (P) is applied by using the largest P_{ij} of all three P matrices and dividing through by P_{ij}. Normalizing all three P matrices provides weighting scheme with a small range of values describing the as the lowest error covariance matrix as the "best" UAV pair. This pair is only for geolocation purposes and not the overall case. A geolocation performance table (see Table 1), based on the normalized data is created to organize the results in useful manner to pass on to the Task Nomination Logic.

Phase (3) Calculations: Fused Geolocation Calculations

After the Task Nomination Logic selects the particular UAV pair to precisely geolocate the emitter in region R1, angle/range to emitter calculations begin as they approach the kill zone. Both UAVs are able to run EKF's in a decentralized manner but one selected as team leader. The team leader initializes its EKF with the available RJ Phase (1) data and if in an emission cycle, fuses both angle/range observations. If not within an emission cycle the leader propagates the state estimate and the error covariance forward in time.

Fusion of the estimate state takes place when the RJ and the UAV team combine their geolocation estimates on board the RJ (i.e., the UAV data are communicated to the RJ throughout the mission). A Global EKF or fusion node based on distributed architecture within the RJ is initialized once the team of UAVs has sent their geolocation result. The UAV team is now a Local [Extended] Kalman Filter (LKF) and the second EKF using angle and range measurements is also a LKF, the third filter is the new Fusion Node (FN).

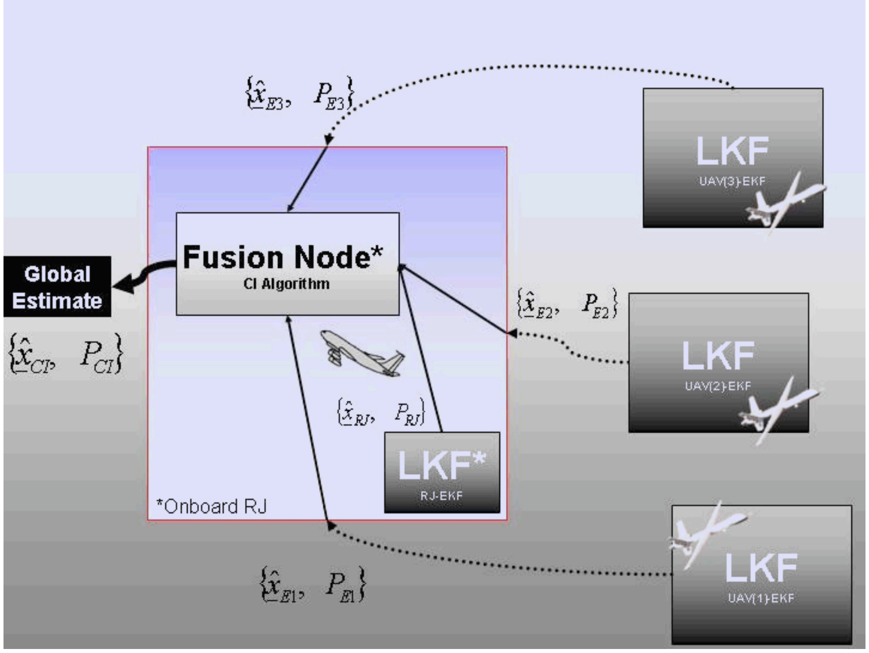
Within this distributed architecture each LKF has a sensor to input measurements. Now in the FN the inputs are taken from the output estimates of the LKF and produce a new estimate based on the fused estimate of each LKF. This architecture is hierarchical fusion without feedback to any of the LKF. An optimal estimate cannot be guaranteed as in the centralized architecture, but near optimal and consistent estimates of the state should be an output from the FN.

As the RJ and UAV combine geolocation estimates their respective geometries vary in time allowing better observations of the emitting target. However at the FN the error covariance may vary in undesirable manner when it receives redundant geolocation estimates. Because of this the FN may become over confident, instead of keeping the same level of confidence for the emitter error covariance. Also, error covariance correlations cannot be optimally accounted for in the LKF or at the FN unlike a centralized EKF.

Applying Covariance Intersection (CI) in the FN meets both requirements of near optimal and consistent geolocation estimates. At the FN each LKF sends their estimate and error covariance matrix for calculations using CI. The CI gives an unbiased estimate that is a linear combination of all information received. Noting that each piece of information has knowledge statistical associated with it we determine the necessary weights for an unbiased estimate. Each piece of information is weighted so the sum of the weights equal

1 and the domain is [0, 1]. Optimizing the weights we can find optimal weights for *N* amounts of information to be fused.

Once the minimal FN error covariance is achieved for weapons strike the RJ, sends the geolocation 3-sigma bounds and estimate to the strike platforms. Fusion-based geolocation is a robust architecture compared to a centralized version. The distributed architecture minimizes single point failures in geolocation calculations while the UAV team and RJ are en-route to separate regions R1 and R2. These operations are depicted in Figure 7 (LKF = Local (Extended) Kalman Filter).



$\{\hat{x}_{E3}, P_{E3}\}$
LKF
UAV(3)-EKF
Fusion Node*
CI Algorithm
Global Estimate
$\{\hat{x}_{CI}, P_{CI}\}$
$\{\hat{x}_{E2}, P_{E2}\}$
LKF
UAV(2)-EKF
$\{\hat{x}_{RJ}, P_{RJ}\}$
LKF*
RJ-EKF
*Onboard RJ
LKF
UAV(1)-EKF
$\{\hat{x}_{E1}, P_{E1}\}$

Figure 7. Operations.

Data Fusion Overview—Level 3 Threat Estimation

The Threats of Concern for the Use Case are those to the two Special Ops Aircraft shown in Figure 7. Since they are observing OpSec and do not have sensors that can detect the pop-up Threat, it can be seen that they may be at risk to that Threat depending on various factors. Using a Threat estimation paradigm that was used on another AFOSR project for Edwards AFB and considered plausible for Joint Strike Fighter basic research applications on that program, we modified that approach for this effort to exploit AFOSR-funded research. That approach involves a logic that estimates the Actual Risk to a friendly aircraft by the relationship between an Inherent Risk and the ability to thwart that risk with available countermeasures. The way this notional process is being implemented is shown in Figure 8.

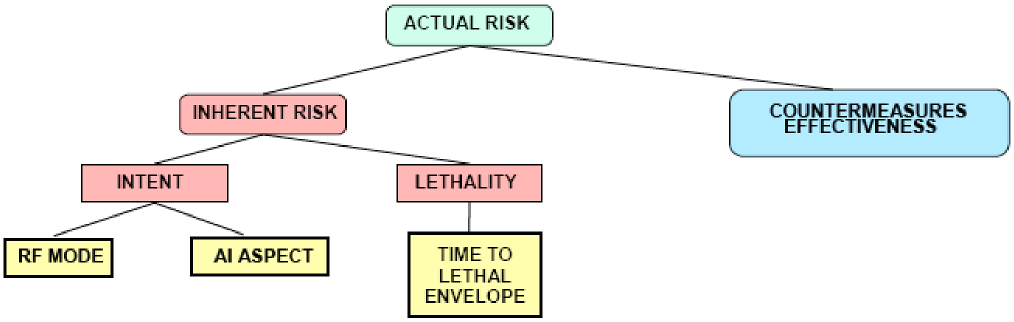


Figure 8. Concept of Risk (~ Threat) Estimation.

The basic idea is for the friendly platform to: (a) sense the hostile emitter modes--here RJ SIGINT, (b) understand the friendly platform-to-hostile threat geometrical aspect (Airborne Intercept Aspect, AI Aspect in the figure)—here this is done by the RJ doing a track estimate for the AC-130 SpOps gunships assumed possible via periodic datalink messages to the RJ and an Extended Kalman Filter on the RJ, and (c) estimate the Time to Lethal Envelope (TTLE) by knowing the AC-130 tracks and the emitter Kill Zone. The Mode and Aspect provide a sense of how much information the hostile may have on the friendlies, and whether the friendlies are in geometry favorable to the hostile—this provides an estimate of Hostile Intent. Lethality, usually interpreted as the ability to do harm, is associated with the TTLE parameter. In turn, Intent and Lethality can be used to develop an estimate of Inherent Risk (meaning the risk present with our consideration of friendly countermeasures). Actual Risk is then the Inherent Risk mitigated by the choice of a possible Countermeasure and its effect on the Threat.

Our approach to these Fusion calculations involved a Fuzzy Logic (FL) technique used on the other AFOSR program, modified for this RJ type application and Use Case. FL methods first “fuzzify” the “crisp” or real-values of pertinent parameters and, using membership functions, represent a degree to which the effects of that parameter influences a consequent result. These dependencies are expressed in Logic Rules that represent the asserted relationships. The composite effect is determined by the applicable set of rules for given parameters and the final result determined by a “defuzzification” step, yielding the Threat or Inherent Risk value as a number in the range (0,1). These Threat/Risk values are used by the Task Nomination logic and DRM process to determine if and when the AC-130 platforms should deviate from their planned courses. The overall Fuzzy Logic process flow is summarized Figure 9.

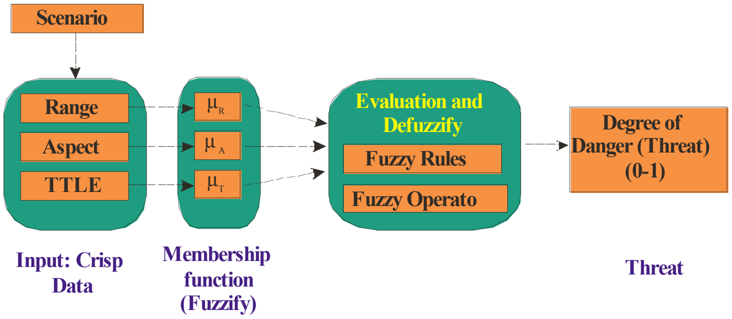


Figure 9. Summary of Fuzzy Logic-based Threat/Inherent Risk Estimation Process.

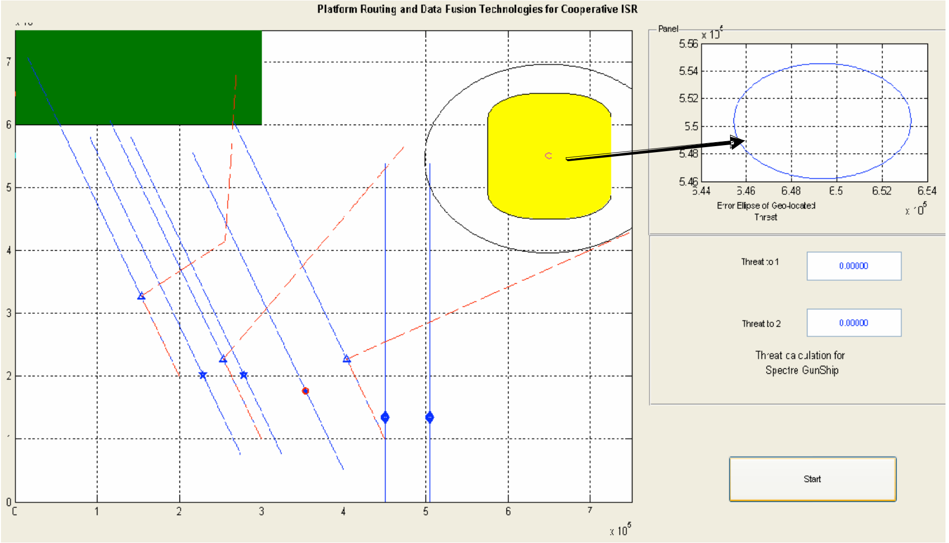
Results

As our effort focused on proof-of-concept, no formal experimentation and analysis was done regarding the results, but clear, quantitative results were realized that gave clear indication of correct trends. The results are shown with screen-shots from our demonstration spawning from the Use Case (see Figures 10 and 11).

Figure 10 illustrates events just after the RJ detects the SIGINT from the pop-up threat, here located in the Yellow region of the figure; the title of the figure elaborates on the circumstances shown. Note the Threat boxes on the middle right side of the screen; no values are computed as yet, since the AC-130 platforms are well out of range. Initially all 3 available UAVs are sent to develop the needed precision geolocation; the screen also shows a later moment after DRM logic has determined that the lower 2 UAVs (Global Hawks) are sufficient for the geolocation task. As a result, the free UAV is sent to re-focus on the second ISR mission.

The second screen (11) illustrates the more precise geo-location in progress using two Global Hawks, with Fusion occurring on the RJ. In the mean time the Fuzzy Logic Threat/Risk calculation is performed for the SpOps gunships as they are advancing towards SAM. The threat values are shown in two text boxes in the right side of the window. Once the threat becomes relatively high, the DRM logic directs Gunship #1 in a new direction to vector away from the kill circle of the SAM. Change in direction is shown with dashed red line. DRM logic also assigns one of the F-22's to fly towards the SAM to provide cover for the SpOps operations. The arrow is pointing towards geo-location error ellipse (Major axis is around 650 meter and minor axis is around 500 meter).

In summary, the composite multi-Level Data Fusion and Dynamic resource Management Logic operations show correct trends and positive results. Precision emitter geolocation is achieved sufficient to provide threat avoidance (note the reduction in error ellipse size from 10km to 650m semi-major axis size).


Platform Routing and Data Fusion Technologies for Cooperative ISR

Threat to 1 0.00000

Threat to 2 0.00000

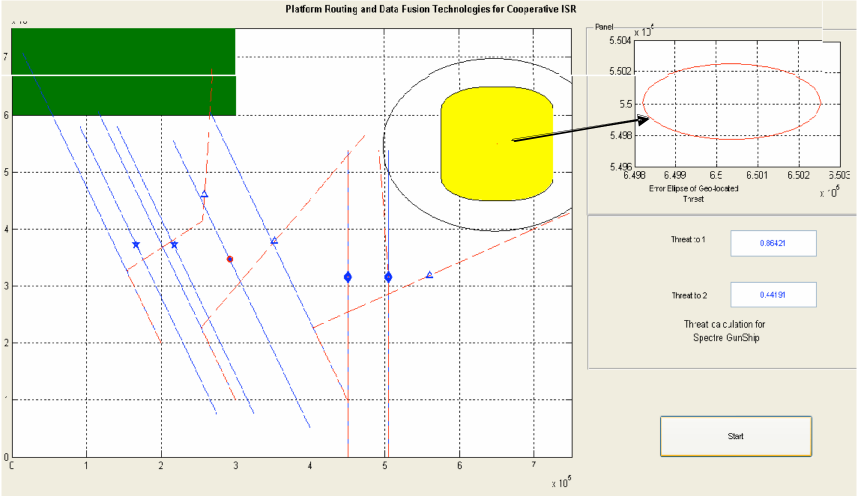
Threat calculation for Spectre GunShip

Start

Legends:

△	**Global Hawk**	●	**Rivet Joint Surveillance Aircraft**
◆	Spectre Gunships	★	F-22

Figure 10. Simulation Scenario when RJ detects hostile emitter signal in yellow region and first Geo-location has been done. Global Hawk Tasks and the path have been modified as per task re-nomination logic. The change Route is shown in dashed red lines. Arrows point towards the error ellipse (blue) for the located SAM (Major axis of ellipse is around 10 kilometers). The black circle is the calculated Kill Zone for the located SAM.


Platform Routing and Data Fusion Technologies for Cooperative ISR
7
6
5
4
3
2
0
C
1
2
3
4
5
6
7
x 10^5
Panel
x 10^5
5.504
5.502
5.5
5.498
5.496
6.498
6.499
6.5
6.501
6.502
6.503
x 10^5
Error Ellipse of Geo-located Threat
Start

Threat to 1 0.86421

Threat to 2 0.44191

Threat calculation for Spectre GunShip

Figure 11. Near-end point of Use Case Simulation. Global Hawk-based Precision Geolocation, Threat Avoidance by AC-130 Gunships.

2.2 TASK 2: DATA ACQUISITION

While our goal was to acquire relevant, “real” RJ operation data used by KAST, we were unable to do so. As a workaround we sought sources that would complement our needs relevant to the stated AFRL RIEA Phase I objectives and our proposed ***fmCortex***^{TM} solution. These included data available from earlier R&D initiatives that team members were involved in that were applicable to those methodologies we were seeking to modify and, or develop. These were consistent with our proposed approach - and relevant to DF and DRM within the context of inter-platform routing and collaborative flow management. An example was an initiative that was sponsored by AFFTC Edwards AFB that focused on Multi-platform Air-to-Air EW Engagement. We were also able to leverage platform routing analyses conducted in like domains and tailor those accordingly.

Our baseline approach and the associated data considered a dynamic adaptation to an existing ATO to deal with contingencies, pop-up threats, etc., while simultaneously finding the ideal means to reassign mission-tasking segments to the entire multi-platform suite. Our Use Case and the resulting methodologies, ontologies and algorithms will lend themselves well to future development in the required domain.

2.3 TASK 3: PLATFORM DYNAMIC RESOURCE ALLOCATION

For this task emphasis was placed on the cooperative dynamic resource allocation requirement for the airborne suite of multiple ISR platforms, addressing the need for a capability to dynamically assign existing and new mission tasking segments to the various platforms as both contingencies and pop-up threat conditions inevitably arise. This was done while simultaneously finding the best way to reassign mission tasking segments to the entire multiplatform suite, using overall mission effectiveness as its central optimization criterion. Our solution is a recommendation to mission staff. [Note: we use the term "recommended" for both the task reassignments and the new routings, since we intend to invoke these capabilities as decision-aids to appropriate mission staff, not as a fully automated capability that we consider unrealistic in the dynamic environments that are encountered in today's operations]. As things are always dynamic, we employed a solution that uses a (possibly-variable) time slice approach in order to deal with contingencies across the mission-planning horizon. [Note too that there is an important *feedback loop* that must be incorporated, so that the reassignment logic is aware of what tasks are being completed or what other contingencies may be being experienced]. We developed a basic dynamic re-tasking model and provided a much more effective and extensible model then initially conceived. Initial mission components only included 'time on target' or 'snapshot' tasks and continuous (e.g. video surveillance) tasks. The model is now extensible to a very wide variety of mission components as detailed in the following section with geolocation tasks and the concept of cover or escort tasks as well as additional constraints on fuel capacity/consumption. The following section provides more details on the mathematical aspects of the solution.

2.4 TASK 4: DEVELOP MATHEMATICAL TECHNIQUES AND ROUTING/CUEING ALGORITHMS

The use of mathematical programming (MP) was the paradigm chosen for optimal resource allocation. Mathematical programming refers to a class of analytical (algebraic) methods that can find the best way to achieve a given objective while complying with a set of constraints. MP models determine the optimal allocation of resources among competing alternatives within an operational system. In addition to the obvious practical benefit of generating optimal solutions, MP provides a sound theoretical basis for properly understanding the broader implications inherent in the framework of the solution structure. For example, the developed models depict the consequences of alternative courses of action by quantifying the opportunity costs of scarce system resources. MP comprises a variety of paradigms (theoretical frameworks) tailored to different kinds of problems, to include: linear programming (LP), integer programming (IP) for problems requiring integer solutions; nonlinear programming (NLP) where the objective and/or one or more constraints are nonlinear functions; and goal programming (GP) for problems with multiple objectives. The *significance* of developing this dynamic resource allocation capability is to off-load the cognitive workload of managing such issues in an optimal way from the flight crews, enable much more rapid response times to determine the reassignments under stressful and possibly life-threatening conditions, while at the same time maintaining a focus on overall mission effectiveness.

In the abstract, parameters and estimates provided by the DF system are among the various criteria used in the Task Nomination Logic Matrix, which is a matrix containing a list of tasking options (typically the list of available/feasible resources), and the various parameters and metrics associated with each resource choice. This matrix is scanned by the DRM logic to assess the best options from the point of view akin to solving an Assignment problem. However, the DRM logic does not take these initial resource assignments as final but weighs them in a separate, mathematical programming logic that formulates a set of complex constraints and objective functions that frame the mission-level optimization approach.

In addition to the time-on-target, snapshot and continuous task types in the earliest model formulation, we added the following additional criteria to the modeling repertoire:

Updates to the Model

- The objective function formulation has been extended to allow the user to provide preferential time slices. For example, in a given time window, it might be preferable to perform the task as early as possible;
- A "moving surveillance" mission component type has been added. These are useful in modeling UAV surveillance where the area of interest is a path between two locations, rather than a single location. Here, two locations on the battlefield are identified, and a single resource is tasked to perform continuous surveillance over the entire path;
- A "geolocation" mission component type has been added. Here, a pair of resources, operating in different locations, is tasked to monitor a remote target whose location needs to be precisely determined. The exact location of each resource, as well as the exact time at which the geolocation task is to be performed, is selected from a list of candidate locations/times;
- The introduction of "fuel capacity" constraints force a resource to return to a base location before the resource runs out of fuel;
- The concept of "support" resources has been included in the model. The support resources are capable of providing coverage for "vulnerable" resources, and are scheduled to fly alongside the vulnerable resources whenever they enter/leave unsafe areas;
- The model notation was updated to handle the new mission component types;
- The model was updated to allow a task to be performed by multiple resources. Previously the model required a task to be assigned to no more than one resource.

The full model is provided on the following pages.

The Full Phase I Mathematical Programming Model for Inter-Platform Re-tasking

Notation - Sets

B	Set of bases (depots).
B_r	Set of bases (depots) that are available to resource r.
M	Set of mission components (tasks). $m \in M$.
$U \subseteq M$	Set of tasks located in unsafe locations.
R	Set of resources.
$V \subseteq R$	Set of vulnerable resources that require support/cover from support resources.
$S_{rv} \subseteq R$	Set of support resources that are capable of providing cover for vulnerable resource $r_v \in V$.
T	Set of time slices making up the mission horizon. $t \in T$.
$T_m \subseteq T$	Set of time slices in which mission component $m \in M$ may be performed.
R_m	Set of resources that can perform mission component $m \in M$. $r \in R_m$.
M_r	Set of mission components (tasks) capable of being performed by resource $r \in R$. $m \in M_r$.
M_{rt}	Set of mission components (tasks) capable of being performed by resource $r \in R$ at time slice $t \in T$. $m \in M_{rt}$.
$\Delta_M^-(r,m)$	Set of *tasks* (excluding bases) from which resource r may travel directly to task m.
$\Delta_B^-(r,m)$	Set of *bases* from which resource r may travel directly to task m. *Tasks* are not included in this set.
$\Delta^-(r,m)$	Set of all possible nodes from which resource r may travel directly to node m. This includes tasks, initial location, and bases.
$\Delta^0(r)$	Initial location of resource r.
$\Delta^+(r)$	Set of all possible nodes to which resource r may travel.
$\Delta_t^+(r)$	Set of all possible nodes to which resource r may travel at time t. This includes tasks and bases, but does not include initial location *unless* the initial location was a base.

Notation - Parameters

0 $s_{rv}^{\min}$	0 Minimum number of support resources that are required for vulnerable resource $r_v \in V$.
0 $s_{rv}^{\max}$	1 Maximum number of support resources that are allowed for vulnerable resource $r_v \in \mathrm{V}$.
0 D_m	2 Number of time slices for which continuous task m must be performed.
0 $n_m^{\min}$	3 Minimum number of resources required to perform task m.
0 $n_m^{\max}$	4 Maximum number of resources allowed to perform task m.
$a_{rm_a m_b}^{t_b} = 1$	If ATO has resource r (initially) assigned to mission component m_b during time slice t_b and the previous mission component was m_a.
$f_{rm_a m_b}$	*Minimum* number of time slices required for resource r to transition from mission component m_a to mission component m_b.

$d_{m_a m_b}$	Euclidean distance between mission components m_a and m_b. Used in objective function term Z_3. $d_{m_a m_a} = 0$.
p_m	Priority of mission *m*.
e_{rm}	Effectiveness of resource *r* conducting mission *m*.

Notation - Decision Variables

$x_{rm_a m_b}^{t_b} = 1$	If resource *r* is to be assigned to mission component m_b during time slice t_b and the previous mission component was m_a.
y_m	Number of \`infinite' resources assigned to perform mission-critical *snapshot* task *m*.
z_m^t	Number of \`infinite' resources assigned to perform mission-critical *continuous* task *m* at time slice *t*.
c_{ab}^t	Number of \`infinite' resources required to provide cover for vulnerable resources on the link between tasks *a* and *b*, where at least one of these tasks is in an unsafe location, *U*.

Objective Functions

$$Z_1 \equiv \sum_{m_b \in M} \sum_{r \in R_{m_b}} p_{m_b} e_{rm_b} \sum_{t_b \in T_{m_b}} \sum_{m_a \in \Delta^-(r,m_b)} (1 - \xi_{m_b} \mid g_{m_b} - t_b \mid) x_{rm_a m_b}^{t_b}$$

Maximize mission priority and resource effectiveness while considering preferences for assigning tasks to target time slices.

$$Z_{2a} \equiv -\sum_{m_b \in A} \sum_{r \in R_{m_b}} \sum_{m_a \in \Delta^-(r,m_b)} \left[\left(1 - \sum_{t_b \in T_{m_b}} a_{rm_a m_b}^{t_b} \right) \sum_{t_b \in T_{m_b}} x_{rm_a m_b}^{t_b} \right]$$

Impose a penalty if *snapshot* tasks are re-assigned to a different resource.

$$Z_{2b} \equiv -\sum_{m_b \in B} \sum_{r \in R_{m_b}} \sum_{m_a \in \Delta^-(r,m_b)} \sum_{t_b \in T_{m_b}} \left[\frac{1}{|T_{m_b}|} \left(1 - a_{rm_a m_b}^{t_b} \right) x_{rm_a m_b}^{t_b} \right]$$

Impose a penalty if *video* tasks are re-assigned to a different resource.

$$Z_{2c} \equiv -\sum_{m_b \in A} \sum_{r \in R_{m_b}} \sum_{m_a \in \Delta^-(r,m_b)} \left[\sum_{t_{b1} \in T_{m_b}} a_{rm_a m_b}^{t_{b1}} \right] \sum_{t_{b2} \in T_{m_b}} \sum_{\substack{t_{b3} \in T_{m_b} \\ t_{b3} \neq t_{b2}}} \left[\left(1 - a_{rm_a m_b}^{t_{b2}} \right) x_{rm_a m_b}^{t_{b3}} \right]$$

Impose a penalty for changing the assigned completion time for *snapshot* tasks.

$$Z_3 \equiv -\sum_{m_b \in M} \sum_{r \in R_{m_b}} \sum_{\substack{m_a \in \Delta^-(r,m_b) \\ m_a \neq m_b}} d_{m_a m_b} \sum_{t_b \in T_{m_b}} x^{t_b}_{r m_a m_b}$$

Minimize the distance traveled for each resource.

$$Z_4 \equiv -\sum_{m \in C} y_m - \sum_{m \in D} \sum_{t \in T_m} z^t_m - \sum_{m \in E} \sum_{t \in T_m} c^t_m$$

Minimize the use of “infinite resources”.

Computations for scaling objective function:

$$Z_1 \le Z_1^{max} \equiv |M| \left(\max_{r,m} \left(p_{m_b}\right)\left(e_{rm_b}\right) \right)$$

$$Z_{2a} + Z_{2b} \le |M|, Z_{2c} \le |MCT \cup MC \cup NMC|$$

If $Z_2 = Z_{2a} + Z_{2b}$ or if $Z_2 = \mathring{a}(Z_{2a} + Z_{2b}) + (1 - \mathring{a})Z_{2c}$, then $Z_2 \le Z_2^{max} \equiv |M|$, where $0 \le \mathring{a} \le 1$.

$$Z_3 \le Z_3^{max} \equiv |M| \left(\max_{m_a \in M, m_b \in M} d_{m_a m_b} \right)$$

Some possible objective function formulations include:

$$\text{Max} \quad \acute{a} \frac{Z_1}{Z_1^{max}} + (1 - \acute{a}) \frac{Z_{2a} + Z_{2b}}{Z_2^{max}} + \hat{a} \frac{Z_3}{Z_3^{max}} + \tilde{a} Z_4,$$

$$\text{Max} \quad \acute{a} \frac{Z_1}{\max_{r,m} p_m e_{rm}} + (1 - \acute{a})(Z_{2a} + Z_{2b}) + \hat{a} \frac{Z_3}{\max_{m_a \in M, m_b \in M} d_{m_a m_b}} + \tilde{a} Z_4, \text{or}$$

$$\text{Max} \quad \acute{a} \frac{Z_1}{Z_1^{max}} + (1 - \acute{a}) \frac{\mathring{a}(Z_{2a} + Z_{2b}) + (1 - \mathring{a})Z_{2c}}{Z_2^{max}} + \hat{a} \frac{Z_3}{Z_3^{max}} + \tilde{a} Z_4,$$

where $0 \le \acute{a} \le 1, 0 \le \hat{a} \le 1, 0 \le \mathring{a} \le 1$, and $\tilde{a} = \left(\max_{m \in M} p_m \right) * \left(\max_{m \in M, r \in R_m} e_{rm} \right)$.

Type I Constraints

Type I constraints describe requirements for performing a given task.

MCA Tasks

MCA tasks are mission-critical tasks to which resources must be assigned at any time in T_m.

$$\sum_{r\in R_m}\sum_{t_b\in T_{m_b}}\sum_{ma\in\Delta^-(r,m_b)} x^{t_b}_{rm_a}m_b + y_{m_b} \le n_{m_b}{}^{\max} \quad \forall m_b \in MCA$$

$$\sum_{r\in R_m}\sum_{t_b\in T_{m_b}}\sum_{ma\in\Delta^-(r,m_b)} x^{t_b}_{rm_a}m_b + y_{m_b} \ge n_{m_b}{}^{\min} \quad \forall m_b \in MCA$$

$$\sum_{t_b\in T_{m_b}}\sum_{ma\in\Delta^-(r,m_b)} x^{t_b}_{rm_a}m_b \le 1 \quad \forall m_b \in \{MCA : n_{m_b}{}^{\max} \ge 2\}, r \in R_{m_b}$$

<u>*MCAL* Tasks</u>

MCAL tasks require that a resource will be busy in transit from task m_b to task m_c. *MCAL* tasks may be used to model a video surveillance task in which video is captured over a path (rather than at a single location).

$$\sum_{r\in(R_{m_b}\cap R_{m_c})}\sum_{t_b\in T_{m_b}}\sum_{ma\in\Delta^-(r,m_b)} x^{t_b}_{rm_a}m_b + y_{m_b} \le n_{m_b}{}^{\max} \quad \forall (m_b,m_c)\in MCAL$$

$$\sum_{r\in(R_{m_b}\cap R_{m_c})}\sum_{t_b\in T_{m_b}}\sum_{ma\in\Delta^-(r,m_b)} x^{t_b}_{rm_a}m_b + y_{m_b} \ge n_{m_b}{}^{\min} \quad \forall (m_b,m_c)\in MCAL$$

$$\sum_{r\in(R_{m_b}\cap R_{m_c})}\sum_{t_c\in T_{m_c}} x^{t_c}_{rm_b}m_c + y_{m_c} \le n_{m_c}{}^{\max} \quad \forall (m_b,m_c)\in MCAL$$

$$\sum_{r\in(R_{m_b}\cap R_{m_c})}\sum_{t_c\in T_{m_c}} x^{t_c}_{rm_b}m_c + y_{m_c} \ge n_{m_c}{}^{\min} \quad \forall (m_b,m_c)\in MCAL$$

$$\sum_{t_c\in T_{m_c}} x^{t_c}_{rm_b}m_c = \sum_{m_a\in\Delta^-(r,m_b)}\sum_{t_c\in T_{m_c}} x^{t_c}_{rm_b}m_c \quad \forall (m_b,m_c)\in MCAL, r\in(R_{m_b}\cap R_{m_c})$$

<u>*MCG* Tasks</u>

MCG tasks are mission-critical geolocation tasks in which exactly two resources are dispatched to two different locations at the same time.

<u>*NMC* Tasks</u>

NMC tasks are not required, but may be useful/beneficial.

$$\sum_{r\in R_m}\sum_{t_b\in T_{m_b}}\sum_{ma\in\Delta^-(r,m_b)} x^{t_b}_{rm_a}m_b \le 1 \quad \forall m_b \in NMC$$

<u>*CMC* Tasks</u>

CMC tasks are mission-critical tasks that must be performed over a continuous set of time slices.

$$\sum_{r\in R_m}\sum_{t_b\in T_{m_b}}\sum_{ma\in\Delta^-(r,m_b)} x^{t_b}_{rm_a}m_b + z^{t_b}_{m_b} \le D_{m_b} \quad \forall m_b \in CMC$$

<u>*CNC* Tasks</u>

Continuous tasks that are not critical but may be useful.

$$\sum_{r\in R_{m_b}} \sum_{m_a\in\Delta^-(r,m_b)} x^{t_b}_{rm_am_b} \le 1 \quad \forall m_b \in CNC, t_b \in T_{m_b}$$

Type II Constraints

Type II constraints ensure that each resource is assigned to no more than one task at a time, according to its capabilities.

$$\sum_{m_b\in\Delta^+{}_{t_b}(r)} \sum_{m_a\in\Delta^-(r,m_b)} x^{t_b}_{rm_am_b} \le 1 \quad \forall r \in R, t_b \in T \qquad (2.1)$$

Type III Constraints

Type III constraints represent time/space constraints. Let $f_{rm_1m_2}$ represent the *minimum* duration (number of time slices) required for resource *r* to move from mission component m_1 to mission component m_2. Each resource requires a single "minimum duration table", resulting in |R| tables. Each table is a square matrix containing ($|M_r|$ + 1) rows/columns; $|M_r|$ mission components for which resource *r* is capable plus one "dummy" mission indicating the initial location of resource *r*.

The following logic is used to construct a set of constraints for *infeasible* mission pairs:

```
foreach resource $r \in R$ {
    foreach mission component $m_c \in \Delta^+(r)$ {
        foreach time slice $t_c \in T_{m_c}$ {
            foreach mission component $m_b \in \{\Delta^-(r,m_c) \setminus \Delta^0(r) : m_b \ne m_c\}$ {
                foreach $\{t_b \in T_{m_b} : t_b \le t_c\} \cup t_0$ {
                    $d = t_c - t_b$
                    if $f_{rm_bm_c} > d$ {
                        $x^{t_c}_{rm_bm_c} \le 1 - \sum_{m_a\in\Delta^0(r,m_b)} x^{t_b}_{rm_am_b}$          (3.1)
```

Type IV Constraints

Type IV constraints specify the network structure.

- Each task must be linked to a predecessor task.

$$\sum_{m_a\in\Delta^-(r,mb)} \sum_{\substack{t_b\in T_{m_b} \\ t_b + f_{rm_bm_c} \le t_c}} x^{t_b}_{rm_am_b} \ge x^{t_c}_{rm_bm_c} \qquad \begin{array}{l} \forall r \in R, m_c \in \Delta^+(r), \\ m_b \in \{\Delta^-(r,m_c) \setminus (\Delta^0(r) \cap \Delta^-{}_M(r,m_c))\}, \\ t_c \in T_{m_c} \end{array}$$

- Each *snapshot* task may have no more than 1 successor tasks for a given resource.

$$\sum_{m_b \in \{\Delta^+(r): m_a \in \Delta^-(r,m_b)\}} \sum_{t_b \in T_{m_b}} x_{rm_a m_b}^{t_b} \leq 1 \quad \forall r \in R, m_a \in (M_r^{snap} \cup \Delta^0(r))$$

- Do not allow resource to "split" on continuous tasks.

$$\sum_{t_b \in \{T_{m_b} : t_b \leq t^{\max}\}} \sum_{m_a \in \Delta^-(r,m_b)} x_{rm_a m_b}^{t_b} \geq \sum_{m_b \in \Delta^+(r,m_b)} \sum_{t_c \in T_{m_c}} x_{rm_b m_c}^{t_c} \quad \forall r \in R, m_a \in M_r^{cont}, t^{\max} \in T$$

- Do not allow resource to "split" at a base location.

$$\sum_{t_b \in \{T_{m_b} : t_b \leq t^{\max}\}} \sum_{m_a \in \Delta^-(r,m_b)} x_{rm_a m_b}^{t_b} + I(\Delta^0(r), m_b) \geq \sum_{m_b \in \Delta^+(r,m_b)} \sum_{t_c \in T_{m_c}} x_{rm_b m_c}^{t_c} \quad \forall r \in R, m_a \in M_r^{cont}, t^{\max} \in T$$

- If a resource is used, it must start from its initial location.

$$B \sum_{m_b \in \{\Delta^+(r): \Delta^0(r) \in \Delta^-(r,m_b)\}} \sum_{t_b \in T_{m_b}} x_{r\Delta^0(r) m_b}^{t_b} \geq \sum_{m_c \in \Delta^+(r)} \sum_{m_b \in \Delta^-(r.m_c)} \sum_{t_c \in T_{m_c}} x_{rm_b m_c}^{t_c} \quad \forall r \in R$$

Note that B is a "big" number. $B=|M||T|$ will suffice.

2.5 TASK 5: DEVELOP OPTIMAL DATA FUSION APPROACHES, METHODOLOGIES

As detailed and depicted in the Use Case under Task 1, consideration was that fusion took place "on" the RJ, and involved algorithms that estimate fusion based threat and risk estimation derived from:

- A Fuzzy Logic Risk calculation;
- RJ and multi-UAV, fusion-based Geolocation algorithms.

The Use Case involved a modern-day mission concept where ISR assets indirectly supported an airborne Special Operations mission. In the scenario was critical that the location of a threat air defense emitter be precisely geo-located in order to provide assured safety to the airborne SOF platforms.

As a result, the focus was on providing two approaches to geolocation, one involving a Kalman Filter-based method and the other an angle/range-based method. In addition, we employed, a Fuzzy Logic-based threat estimation logic that employed the kinematic and geometric estimates provided by the geolocation calculations to estimate the risk to the SOF platforms (see Figure 12). This risk calculation was employed by the dynamic resource allocation logic to optimize the reassignment of ISR assets for SOF protection.

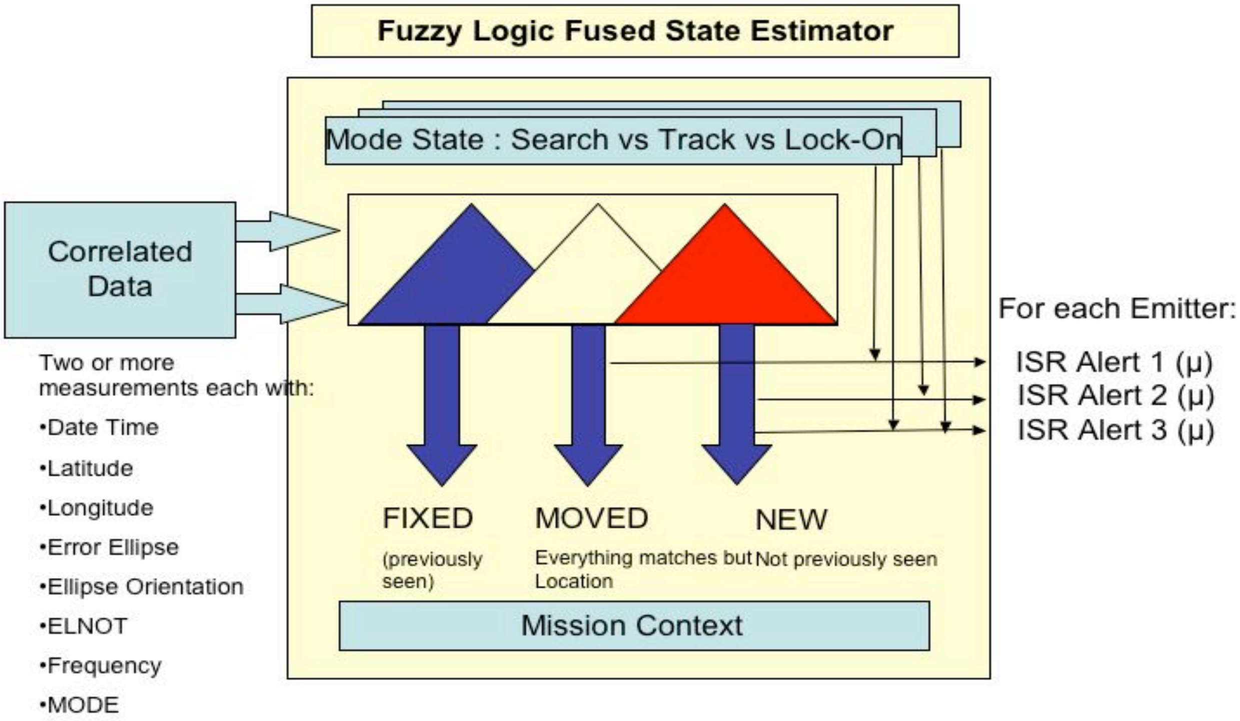
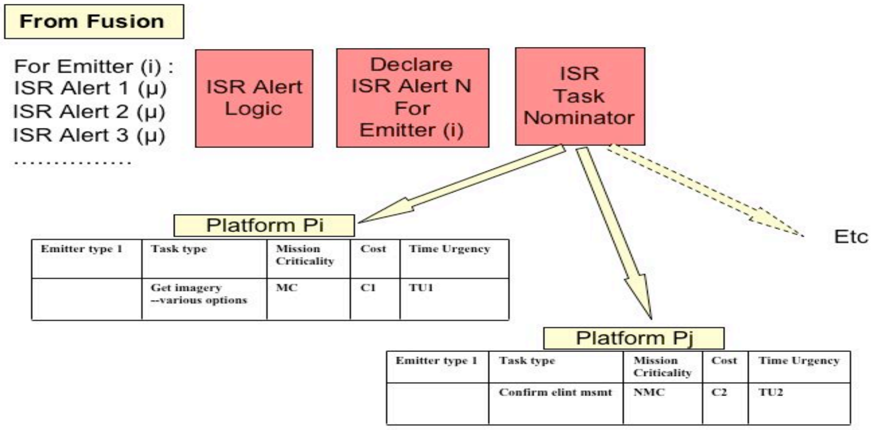


Figure 12. Notional Fused State Emitter Estimate.

Thus, consistent with the design philosophy described above, what would be develop is a cooperative ISR, closed-loop process involving two fusion-based functions (geolocation and threat estimation) and a robust dynamic platform and functional re-tasking logic that is based on optimization techniques employing mathematical programming (see Figure 13).



Emitter type 1	Task type	Mission Criticality	Cost	Time Urgency
	Get imagery --various options	MC	C1	TU1

Emitter type 1	Task type	Mission Criticality	Cost	Time Urgency
	Confirm elint msmt	NMC	C2	TU2

Figure 13. ISR Alert to Tasking Logic
(Note: this is new and needs to be developed).

2.6 TASK 6: DEFINE AND DEVELOP THE FLOW MANAGEMENT CORE ARCHITECTURE (*FMCORTEX*^{TM})

While were unable to gain access to any of the metadata used by the KAST software relative to the RJ – which would have enabled us to make even further progress, we were able to define (and *refine*) an extensible architecture in the context of what was initially proposed (Note: emphasis is on the word refine in that what is shown in Figure XX, is a more tailored framework based on the DF and DRM capabilities developed). The core architecture represents those functions described prior and would be employed as flexible, extensible architecture with the ability to combine processing and parameter settings on the fly, predict and examine the quality of the results, and estimate the processing time and resources required for the intended processes. With those requirements in mind, several important design objectives became apparent, including:

- Easy to add new tasking, logic, and processes;
- Support multiple levels of processing for each task;
- Automated as possible;
- Support multiple data sources;
- Easy to add support for new data sources;
- Management of large numbers of data sources;
- Self-describing data;
- Audit trails of the processing done to the data; and,
- Attention to distributed processing objectives.

Collectively, the ***fmCortex***^{TM} is comprised of those fusion and dynamic resource functions created for or used in the context of our solution. These form the basis of a core capability that when brought into the ***fmCortex***^{TM} architecture yield results consistent with AFOSR and RIEA needs. The following (Figure 14) details the ***fmCortex***^{TM} architecture, individual components and processes. The components and processes are described in the subsections that follow.

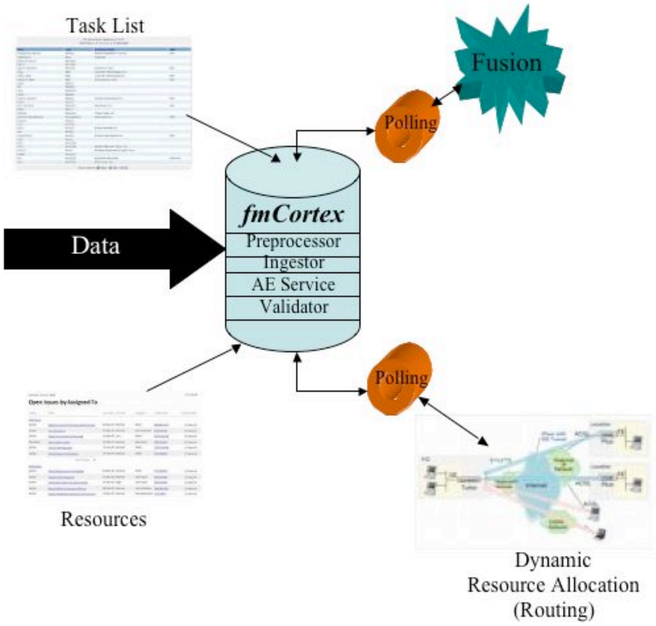


Figure 14. ***fmCortex***^{TM} Core Architecture.

2.6.1 Ingestor

The Ingestor performs the necessary task of bringing the data from the bus and preparing it for use in the ***fmCortex***^{TM} system. The input source data must be carefully examined and tagged in order to be processed efficiently by the Fusion, Routing, and Analytical Engine modules. Information retrieved for tagging purposes includes (but is not limited to): geolocation, error models, time-tags, INT-ID, threat ID, etc. In addition, incoming data must be validated with existing accepted data and models to determine its consistency and/or applicability with known information. This Ingestor module will perform the necessary cataloguing and metadata-tagging to extract the relevant information that can be passed along with the raw data to subsequent processing stages of ***fmCortex***^{TM}.

2.6.2 Analytical Engine (AE) Service

The AE service acts as the central distribution hub and decision making center for the ***fmCortex***^{TM} architecture. It handles the data and disseminates it to the Fusion and Routing modules for additional processing. Central to the Analytical Engine is the concept of polling stations. Both the Fusion and Routing modules have polling station access into the Analytical Engine. These polling stations alert the modules when new data is ready to be processed and carry processed data back to the Analytical Engine. Further discussion on the functionality of these polling stations will be given in the next section. The Analytical Engine also contains the core analytical capabilities of ***fmCortex***^{TM}, the logic of which drives the system. This "brain" of the system is intended to be both flexible and extensible, allowing interpretation of a variety of inputs, addition/deletion of input sources on the fly, and extensible to provide for future enhancements to the logic system.

2.6.3 Validator

The Validator consists of a combination of automated and semi-automated methods to evaluate the results from the Fusion and Analytical Engine modules. The validation process is founded on the triple concepts of accuracy, consistency, and usability. Validation of decisions and fusion results are both data- and task-based. Data based validation is intended to answer the questions: "is the data/model consistent with past models" and "is the data/model accurate". Task based validation is centered on the question "is the data/model sufficient to proceed with the designated task". Validation is used as a control in the analytical process. The control is as follows: are the intelligence requirements for the given task sufficient? If "yes", proceed with the given task. If "no", proceed to data request (e.g., request to the Router for tasking a data collection platform to fill in the gaps in our knowledge).

The overarching objective to our approach is to employ a Service Oriented Architecture (SOA) based approach by enabling system-level architects to describe functionality in multiple processor systems as if it were implemented on a single processor. This provides greater architectural flexibility, especially since functionality is never locked into a particular implementation. The SOA approach was developed to offer a common communication framework between distributed software components in a system – in our case, ***fmCortex***^{TM}. SOA architectures allow one to abstract the processing functions, the data server, and the interfaces used to communicate between them, enabling a solution that is both modular, and readily extensible without need for significant architecture changes between software updates.

2.7 TASK 7: PHASE II PROTOTYPING DESIGN

After completing each of the tasks leading up to the prototype design it was determined to modify the original design of ***fmCortex***^{TM} to produce an even better streamlined, modular, and extensible product as illustrated in the previous figure. The functionalities similar to those presented in the original proposal, however, as greater functionality was developed under the Phase I than what was originally proposed (and anticipated), coupled with an emphasis on an inter-platform approach, we feel that any minor modifications are enhancements - as well as being more streamlined for internal efficiency.

2.8 TASK 8: MARKET STUDY (*OPTION*)

Although the option was not exercised, (we were scheduled to begin our market study during month 7 of the base contract) our objective, and thus our approach throughout the Phase 1, was to lay the foundation for a capability that would initially lend itself to the AFRL RIEA CMS/KAST program. Additionally, our underlying approach throughout the development process is to continuously maintain consideration for marketing opportunities. What follows are just a few of the potential markets (by function) where an ***fmCortex***^{TM} solution could be brought to bear:

- Supply chain: Defense (to include a myriad of contractors), Transportation, Vendors (e.g., E2open, Boeing, Hitachi, IBM, LG, Motorola, Exostar, BAE, Lockheed Martin, Raytheon, etc.) (there are many others);

- Logistics: Aerospace and Defense, Transportation, Cargo Owners, Manufacturing, Retail;

- Multi-sensor and Data fusion: Aerospace and Defense (to include a myriad of contractors), DHS, Law Enforcement, Data Warehousing, Medical, etc. (again there are many others);

- Dynamic Resource Management: Aerospace and Defense (to include a myriad of contractors), Pharmaceutical, Medical, Application Developers (e.g., Planisware, Serena Mariner, etc.);

- Collaborative Flow: Aerospace and Defense, FAA, Pharmaceutical, Medical, Application Vendors (e.g., 3G Interactive, Fujitsu Software Corp, IBM, Imagesoft, Taligent, etc.).

This large customer base continues to reinforce our determination for pursuing this transition path. While our list includes potential competitors, e.g., Defense contractors, we foresee great potential in either integrating the ***fmCortex***^{TM} product into their environment or them purchasing the software to perform the integration.

3 CONCLUSIONS

We have accomplished each of our Phase I tasks, exceeding our original goals, and are well suited to begin Phase II development.

Our approach was the development of a Data Fusion, Dynamic Resource Management, and Platform Routing (PR) within a Collaborative Flow Management system having multiple capabilities, each converging toward the collective advanced support of the Rivet Joint – through the CMS/KAST initiative. We described in detail the tools required to lay the foundation of the ***fmCortex***^{TM} solution including: pre- and post-processing issues critical for successful fusion, validation, data fusion in the context of cooperative ISR, standard-product assessment, entity/event/alert generation and re-generation, dynamic tasking, etc., as well as dynamic database components.

Our Phase I efforts in DF, DRM and PR, focused on a mix of these categories of applications, enabling us to explore and develop proof-of-concept technical capabilities addressing exemplar problems in reflecting the need to balance resource demands across composite mission operations.

The major conclusions from our work:

- A layered Level 1 Precision Geolocation framework and algorithm-set was prototyped and successfully demonstrated;
- A Fuzzy Logic Level 3 Threat Estimation fusion logic was prototyped and successfully demonstrated;
- Preliminary Task Nomination Logic was developed and integrated into the adaptive DF-DRM process;
- A robust, computationally-efficient Mathematical Programming approach to DRM was designed, modified, and successfully demonstrated;
- These technology prototypes have shown good promise for Cooperative ISR through a synergistic interoperation between Data Fusion and Dynamic Resource Management quantitative techniques;
- An open, modular, extensible, and more efficient architecture was designed and is poised for future development.